



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/678,263

10/06/2003

Kevin David Safford

10992268-3

3447

22879

7590

12/05/2006

HEWLETT PACKARD COMPANY  
P O BOX 272400, 3404 E. HARMONY ROAD  
INTELLECTUAL PROPERTY ADMINISTRATION  
FORT COLLINS, CO 80527-2400

EXAMINER

MEONSKE, TONIA L

ART UNIT

PAPER NUMBER

2181

DATE MAILED: 12/05/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b> 10/678,263	<b>Applicant(s)</b> SAFFORD ET AL.	
	<b>Examiner</b> Tonia L. Meonske	<b>Art Unit</b> 2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 06 October 2003.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 21-40 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 21-40 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 06 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

## DETAILED ACTION

### *Priority*

1. Applicant's claim for the benefit of a prior-filed application under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, or 365(c) is acknowledged. Applicant has not complied with one or more conditions for receiving the benefit of an earlier filing date under 35 U.S.C. 120 as follows:
2. If applicant desires to claim the benefit of a prior-filed application under 35 U.S.C. 120, a specific reference to the prior-filed application in compliance with 37 CFR 1.78(a) must be included in the first sentence(s) of the specification following the title or in an application data sheet. For benefit claims under 35 U.S.C. 120, 121 or 365(c), the reference must include the relationship (i.e., continuation, divisional, or continuation-in-part) of the applications.
3. If the instant application is a utility or plant application filed under 35 U.S.C. 111(a) on or after November 29, 2000, the specific reference must be submitted during the pendency of the application and within the later of four months from the actual filing date of the application or sixteen months from the filing date of the prior application. If the application is a utility or plant application which entered the national stage from an international application filed on or after November 29, 2000, after compliance with 35 U.S.C. 371, the specific reference must be submitted during the pendency of the application and within the later of four months from the date on which the national stage commenced under 35 U.S.C. 371(b) or (f) or sixteen months from the filing date of the prior application. See 37 CFR 1.78(a)(2)(ii) and (a)(5)(ii). This time period is not

extendable and a failure to submit the reference required by 35 U.S.C. 119(e) and/or 120, where applicable, within this time period is considered a waiver of any benefit of such prior application(s) under 35 U.S.C. 119(e), 120, 121 and 365(c). A benefit claim filed after the required time period may be accepted if it is accompanied by a grantable petition to accept an unintentionally delayed benefit claim under 35 U.S.C. 119(e), 120, 121 and 365(c). The petition must be accompanied by (1) the reference required by 35 U.S.C. 120 or 119(e) and 37 CFR 1.78(a)(2) or (a)(5) to the prior application (unless previously submitted), (2) a surcharge under 37 CFR 1.17(t), and (3) a statement that the entire delay between the date the claim was due under 37 CFR 1.78(a)(2) or (a)(5) and the date the claim was filed was unintentional. The Director may require additional information where there is a question whether the delay was unintentional. The petition should be addressed to: Mail Stop Petition, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

4. If the reference to the prior application was previously submitted within the time period set forth in 37 CFR 1.78(a), but not in the first sentence(s) of the specification or an application data sheet (ADS) as required by 37 CFR 1.78(a) (e.g., if the reference was submitted in an oath or declaration or the application transmittal letter); and the information concerning the benefit claim was recognized by the Office as shown by its inclusion on the first filing receipt, the petition under 37 CFR 1.78(a) and the surcharge under 37 CFR 1.17(t) are not required. Applicant is still required to submit the reference in compliance with 37 CFR 1.78(a) by filing an amendment to the first sentence(s) of the specification or an ADS. See MPEP § 201.11.

***Specification***

5. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.
6. The disclosure is objected to because of the following informalities:
  - a. On page 6, line 2, please change "90" to -90a and 90b--.
7. Appropriate correction is required.

***Claim Objections***

8. Claims 23, 33, 21 and 38 are objected to because of the following informalities:
  - a. In claim 23, line 2 and in claim 33, line 2, the limitation "QNaN" is an acronym and the words it represents should be spelled out in the claims the first time it is used in each claim.
  - b. In claim 21, line 10, and in claim 38, line 1, the limitation "ROM" is an acronym and the words it represents should be spelled out in the claims the first time it is used in each claim.
9. Appropriate correction is required.

***Claim Rejections - 35 USC § 112***

10. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
11. Claims 24, 26 and 32-40 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

12. Referring to claim 32, the limitation "Substantially Simultaneously switch contents" in line 4 is unclear. The scope of the claim cannot be determined in light of the specification. The specification has not set forth a clear standard for measuring what "substantially simultaneously" means. See MPEP 2173.05 regarding relative terms of degree. On page 7 of the specification, Applicant has made references to "substantially at the same time" which is the same as "substantially simultaneously", but the term is used in different contexts with different standards and examples of what it might mean in different situations. For example, page 7, lines 2-5 state: *"(1) read their respective operands at substantially the same time, (2) write their results at substantially the same time (at least one clock cycle after the read), and (3) finish executing their instructions at substantially the same time (i.e., they operate in lockstep)"*. Under item (2) "substantially the same time" appears to mean at least one clock cycle after the read and under item (3) "substantially the same time" appears to mean operating in lockstep. Does "substantially simultaneously switch content" mean switching within a clock cycle or switching in lockstep or something else recognized by one of ordinary skill in the art such as switching in parallel or switching within one or two clock cycles of each other? For purposes of examination "substantially simultaneously switch contents..." is interpreted as to complete switching contents in parallel. Appropriate correction is required.

13. Claims 24, 34 and 39 also contain the limitation "substantially simultaneously..." and are unclear for similar reasons with respect to claim 32 above. With respect to claim 24 "substantially simultaneously" is interpreted to mean that the claimed data

moving to the first register and the claimed data moving to the second register is performed in parallel. Claim 34, line 2, "substantially simultaneously switch contents..." is interpreted as to switch contents in parallel. Claim 39, line 7, "substantially simultaneously switching contents" is interpreted as to switch contents in parallel.

14. Referring to claim 32, the limitation "vice versa" makes the limitation "switch the contents of the top of the register stack with the contents of another register of the register stack and vice versa, such that an error does not occur" unclear. The metes and bounds of the claim cannot be determined. Is there one switch or two switches? Is the content of the top of the register changed once or twice? Is the content of another register changed once or twice? For purposes of examination, the limitation is interpreted as meaning the value on the top of the stack is switched with the value at another register, such that the values are only switched once. Appropriate correction is required.

15. Referring to claim 34, the limitation "vice versa" makes the limitation "switch the contents of the top of the register stack with the contents of another register of the register stack and vice versa" unclear. The metes and bounds of the claim cannot be determined. Is there one switch or two switches? Is the content of the top of the register changed once or twice? Is the content of another register changed once or twice? For purposes of examination, the limitation is interpreted as meaning the value on the top of the stack is switched with the value at another register, such that the values are only switched once. Appropriate correction is required.

16. Referring to claim 39, the limitation "vice versa" makes the limitation "switching contents of the top register of the register stack with contents of another register of the register stack and vice versa, such that an error does not occur" unclear. The metes and bounds of the claim cannot be determined. Is there one switch or two switches? Is the content of the top of the register changed once or twice? Is the content of another register changed once or twice? For purposes of examination, the limitation is interpreted as meaning the value on the top of the stack is switched with the value at another register, such that the values are only switched once. Appropriate correction is required.

17. Claim 26, 33-38 and 40 are rejected for incorporating the defects of the claims from which they depend.

***Claim Rejections - 35 USC § 102***

18. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

19. Claims 21-40 are rejected under 35 U.S.C. 102(b) as being anticipated by Goddard et al., US Patent 5,696,955 (herein referred to as Goddard).

20. Referring to claim 21, Goddard has taught a system comprising:

- a. main memory storing an instruction set (Figure 1A, element 114, column 4, lines 20-29); and

- b. a processor (Figure 1, element 110) operably connected to main memory (Figure 1, element 114) by a bus network (column 4, lines 20-29, element 111), wherein the processor comprises:
- i. a floating point unit (column 6, lines 36-39, column 7, lines 50-52, Figure 1B, at least element 120 and 122, comprise the claimed floating point unit as they both execute floating point instructions in the processor.);
  - ii. a register stack (column 2, line 45-column 3, line 40, floating point stack, column 7, lines 9-15);
  - iii. dependency checking logic for determining whether instructions are executed sequentially or in parallel (column 14, lines 2-10, column 4, lines 50-63, column 5, line 50-column 6, line 12, Up to 5 instructions are issued and executed each clock cycle based on a dependency checking circuit.);
  - iv. two execution units for executing instructions (Figure 1B, at least elements 120, 121, 122 and 180); and
  - v. ROM storing a micro-code handler that is invoked when two move instructions operating on registers in the register stack are executed in parallel and cause a stack underflow exception (column 10, lines 48-53, column 11, line 67-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 32, FXCH is a floating point exchange instruction containing two instructions: an instruction to move a value from the top of the stack to a specific location in the stack and an instruction to move a

value at the specific location in the stack to the top of stack, such that two values in the stack are exchanged. Exchanging values of pointers in the stack is equivalent to moving the values as the pointers determine the location of data values in the stack. When a pointer for a data value changes, then the location for that data value effectively changes. When the two instructions in the FXCH instruction indicate a stack underflow exception, a micro-code handler from the ROM is invoked.), the micro-code handler being operable to overcome any dependency between the registers and allow execution of the move instructions in parallel without a stack underflow exception (column 11, line 57-column 12, line 3, column 16, line 49-column 17, line 35, Upon an error, such as an underflow exception, a resynchronization response restarts the processor to re-execute the FXCH instruction, which includes the move instructions, without an exception.).

21. Referring to claim 22, Goddard has taught the system of claim 21, as described above, and wherein the micro-code handler is operable to flush the two move instructions if any of the two move instructions causes the stack underflow exception (column 3, lines 35-41, column 6, line 59-column 7, line 9, Remaining stages of the operation are cancelled upon an underflow exception. Column 17, lines 46-50).

22. Referring to claim 23, Goddard has taught the system of claim 21, as described above, and wherein the micro-code handler is operable to replace contents of one of the registers with a QNaN in response to detecting the stack underflow exception (column

7, lines 53-64, column 17, lines 15-31, In response to an underflow exception, the resynchronization micro-code replaces the register state value with a QNaN.).

23. Referring to claim 24, Goddard has taught the system of claim 21, as described above, and wherein the execution units are operable to execute the move instructions in parallel such that first data of a first register of the registers is moved to a second register of the registers, and second data of the second register is substantially simultaneously moved to the first register (column 2, lines 8-25, column 7, line 53-column 8, line 7, column 12, lines 43-49, column 20, lines 35-56, When the FXCH instruction issues, then both move instructions that exchange the register stack values issue in parallel.)

24. Referring to claim 25, Goddard has taught the system of claim 21, as described above and wherein the first register is a top register in the floating point stack and the second register is another register in the floating point stack (abstract, column 20, lines 46-56).

25. Referring to claim 26, Goddard has taught the system of claim 24, as described above, and wherein the register stack is for the floating point unit (column 6, lines 36-39, column 7, lines 50-52, Figure 1B, at least element 120 and 122, The register stack is for the branch unit, where the branch unit is considered part of the floating point unit since the branch unit executes some floating point instructions. Therefore the register stack is for the floating point unit.).

26. Referring to claim 27, Goddard has taught in a processor based computer system having dependency checking logic (column 14, lines 2-10, column 4, lines 50-

Art Unit: 2181

63, column 5, line 50-column 6, line 12, Up to 5 instructions are issued and executed each clock cycle based on a dependency checking logic.) and a register stack (column 2, line 45-column 3, line 40, floating point stack, column 7, lines 9-15), a method comprising:

a. overriding the dependency logic such that move instructions associated with the stack registers are operable to be executed in parallel (column 7, lines 43-64, column 10, lines 48-53, column 11, line 67-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 32, FXCH is a floating point exchange instruction containing two instructions: an instruction to move a value from the top of the stack to a specific location in the stack and an instruction to move a value at the specific location in the stack to the top of stack, such that two values in the stack are exchanged. Exchanging values of pointers in the stack is equivalent to moving the values as the pointers determine the location of data values in the stack. When a pointer for a data value changes, then the location for that data value effectively changes. The branch unit overrides the dependency logic by using the remap stack to speculatively execute FXCH instructions.);

b. executing the move instructions in parallel (column 10, lines 48-53, column 11, line 67-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 32, FXCH is a floating point exchange instruction containing two instructions that execute in parallel: an instruction to move a value from the top of the stack to a specific location in the stack and an instruction to move a value at

the specific location in the stack to the top of stack, such that two values in the stack are exchanged at the same time. Exchanging values of pointers in the stack is equivalent to moving the values as the pointers determine the location of data values in the stack.);

c. determining whether a stack underflow exception has occurred (column 16, lines 57-61) and if it has:

d. flushing the move instructions (column 3, lines 35-41, column 6, line 59-column 7, line 9, column 16, line 49-column 17, line 31, Column 17, lines 46-50, Remaining stages of the operation are cancelled upon an underflow exception such that the move instructions are flushed.); and

e. invoking a micro-code handler algorithm that operates to allow execution of the move instructions in parallel without a stack underflow exception (column 10, lines 48-53, column 11, line 57-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 35, Upon an error, such as an underflow exception, a resynchronization response restarts the processor to re-execute the FXCH instruction with a micro-code handler, which includes the move instructions, without an exception.).

27. Referring to claim 28, Goddard has taught the method of claim 27, wherein the register stack comprises a top register and a selectable register (abstract, column 20, lines 46-56), and wherein invoking the microcode handler further comprises:

- a. determining whether the top register of the stack is empty (column 7, lines 53-64, column 17, lines 15-31, An underflow condition is detected which determines that the top register is empty.); and
  - b. replacing contents of the top register with a defined architectural response in response to said top register being empty (column 7, lines 53-64, column 17, lines 15-31, In response to an underflow exception, the resynchronization micro-code replaces the register state value with a QNaN or valid data.).
28. Referring to claim 29, Goddard has taught the method of claim 28, as described above, and wherein determining whether the top register of the stack is empty, further comprises:
- a. replacing the contents of the selectable register with the defined architectural response in response to the top register of the stack being not empty (column 7, lines 53-64, column 17, lines 15-31, The registers are replaced with a QNaN or valid data.).
29. Referring to claim 30, Goddard has taught the method of claim 29, as described above and further comprising:
- a. executing the move instructions in parallel if the top or selectable register contents have to be replaced with the defined architectural response (column 11, line 57-column 12, line 3, column 16, line 49-column 17, line 35, Upon an error, such as an underflow exception, a resynchronization response restarts the processor to re-execute the FXCH instruction, which includes the two parallel move instructions, and replace the registers with valid data or a QNaN.).

30. Referring to claim 31, Goddard has taught the method of claim 28, as described above and further comprising:

a. replacing the contents of the selectable register with the defined architectural response if the contents of the top register have been replaced with the defined architectural response and a stack underflow exception has occurred (column 16, line 49-column 17, line 35, Data in the register stack is replaced with an architectural response, either valid data or QNaN, in both the top of stack and the selectable register upon an underflow exception.).

31. Referring to claim 32, Goddard has taught a processor, comprising:

a. dependency checking logic (column 14, lines 2-10, column 4, lines 50-63, column 5, line 50-column 6, line 12, Up to 5 instructions are issued each clock cycle based on dependency checking logic.); and

b. a register stack having a top register and a plurality of other registers (abstract, column 20, lines 46-56); wherein the processor is configured to:

c. override the dependency logic such that operations related to the register stack are operable to be executed in parallel (column 7, lines 43-64, column 10, lines 48-53, column 11, line 67-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 32, FXCH is a floating point exchange instruction containing two parallel instructions: an instruction to move a value from the top of the stack to a specific location in the stack and an instruction to move a value at the specific location in the stack to the top of stack, such that two values in the stack are exchanged. Exchanging values of pointers in the stack is equivalent to

moving the values as the pointers determine the location of data values in the stack. When a pointer for a data value changes, then the location for that data value effectively changes. The branch unit overrides the dependency logic by using the remap stack to speculatively execute FXCH instructions.);

d. substantially simultaneously switch contents of the top register of the register stack with contents of another register of the register stack and vice versa, such that an error does not occur (column 10, lines 48-53, column 11, line 67-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 32, FXCH is a floating point exchange instruction containing two instructions that execute in parallel: an instruction to move a value from the top of the stack to a specific location in the stack and an instruction to move a value at the specific location in the stack to the top of stack, such that two values in the stack are exchanged at the same time. Exchanging values of pointers in the stack is equivalent to moving the values as the pointers determine the location of data values in the stack.); and

e. execute an algorithm that replaces the contents of the top register, the other register, or both the top and other registers with a defined architectural response if an exception occurs when the content of both registers are switched (column 7, lines 53-64, column 17, lines 15-31, Upon an underflow exception the registers are replaced with a QNaN or valid data.).

32. Referring to claim 33, Goddard has taught the processor of claim 32, as described above, and wherein the defined architectural response is a QNaN (column 17, lines 15-31).

33. Referring to claim 34, Goddard has taught the processor of claim 32, as described above and further comprising:

- a. at least two execution units (Figure 1B, at least elements 120, 121, 122 and 180) operable to substantially simultaneously switch the contents of the top register of the register stack with the contents of the another register of the register stack and vice versa (column 10, lines 48-53, column 11, line 67-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 32, FXCH instruction switches the content of floating point stack register values in parallel.).

34. Referring to claim 35, Goddard has taught the processor of claim 34, as described above and wherein the at least two execution units are provided in at least one pipelined circuit (Figure 1B, at least elements 120, 121, 122 and 180. Figure 1A and 1B comprise a pipelined circuit.).

35. Referring to claim 36, Goddard has taught the processor of claim 34, as described above and further comprising a floating point unit (column 6, lines 36-39, column 7, lines 50-52, Figure 1B, at least element 120 and 122, comprise the claimed floating point unit as they both execute floating point instructions in the processor.).

36. Referring to claim 37, Goddard has taught the processor of claim 36, as described above and wherein the register stack stores data for the floating point unit

(column 7, lines 43-64, The floating point unit and the branch utilize data and return results on the stack.).

37. Referring to claim 38, Goddard has taught the processor of claim 32, as described above, and further comprising a ROM storing a micro-code that includes the algorithm that replaces the contents of the top register, the other register, or both registers with a defined architectural response if an exception occurs when the content of both registers are switched (column 7, lines 53-64, column 11, line 67-column 12, line 3, column 17, lines 15-31, Upon an underflow exception from an FXCH instruction, the stack registers are replaced with a QNaN or valid data using instructions from the microcode ROM.).

38. Referring to claim 39, Goddard has taught an apparatus comprising:

- a. at least two execution unit means for executing instructions (Figure 1B, at least elements 120, 121, 122 and 180);
- b. dependency checking means for checking the dependency of data associated with instructions executed by the at least two execution unit means (column 14, lines 2-10, column 4, lines 50-63, column 5, line 50-column 6, line 12, Up to 5 instructions are issued and executed each clock cycle based on a dependency checking circuit and logic.);
- c. register stack means for storing data and including a top register and a plurality of other registers (abstract, column 20, lines 46-56);
- d. switching means for substantially simultaneously switching contents of the top register of the register stack means with contents of another register of the

register stack means and vice versa, such that an error does not occur (column 10, lines 48-53, column 11, line 67-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 32, FXCH instruction switches the content of floating point stack register values in parallel.); and

e. execution means for executing an algorithm that overrides the dependency checking means such that instructions related to the register stack means are operable to be executed in parallel (column 7, lines 43-64, column 10, lines 48-53, column 11, line 67-column 12, line 3, column 12, lines 43-47, column 16, line 49-column 17, line 32, Figure 1B, element 120, FXCH is a floating point exchange instruction containing two instructions: an instruction to move a value from the top of the stack to a specific location in the stack and an instruction to move a value at the specific location in the stack to the top of stack, such that two values in the stack are exchanged. Exchanging values of pointers in the stack is equivalent to moving the values as the pointers determine the location of data values in the stack. When a pointer for a data value changes, then the location for that data value effectively changes. The branch unit overrides the dependency logic by using the remap stack to speculatively execute FXCH instructions.) and that replaces the contents of the top register, the other register, or both the top and other registers with a defined architectural response if an exception occurs when the content of both registers are switched (column 7, lines 53-64, column 17, lines 15-31, Upon an underflow exception from and FXCH instruction, the registers are replaced with a QNaN or valid data.).

39. Referring to claim 40, Goddard has taught the apparatus of claim 39, as described above and further comprising a floating point unit means for executing floating point operations (column 6, lines 36-39, column 7, lines 50-52, Figure 1B, at least element 120 and 122, comprise the claimed floating point unit as they both execute floating point instructions in the processor.) and the register stack means stores data for the floating point operations (column 7, lines 43-64, The floating point unit and the branch unit utilize data and return results on the stack.).

### ***Conclusion***

40. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- a. Goddard et al., U.S. Patent 5,857,089 has taught a Floating point stack and exchange instruction,
- b. Sharangpani, U.S. Patent 5,522,051 has taught parallel exchange instructions,
- c. White et al., U.S. Patent 5,764,938 has taught Resynchronization of a superscalar processor,
- d. Bujanos, U.S. Patent 5,572,664 has taught arithmetic operations acting upon an SNaN cause an invalid-operation exception, unless the instruction is a FXCH (floating point exchange),
- e. Meier et al., U.S. Patent 6,370,637 has taught stack register renaming for register exchange operations,

- f. Dao et al., U.S. Patent 5,991,863 has taught generating register stack addresses in a microprocessor,
- g. Song, U.S. Patent 6,079,011 has taught a stack-top updating unit for processing an exchange instruction and a load instruction in parallel in a pipelined processor having a stack register file, and
- h. Intel "Pentium® Processor for Embedded Applications", December 1998, Chap. 31, pages 31-1 to 31-53.

41. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tonia L. Meonske whose telephone number is (571) 272-4170. The examiner can normally be reached on Monday-Friday with first Friday's off.

42. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Fritz Fleming can be reached on (571) 272-4145. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2181

43. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

tlm

TONIA L. MEONSKE

*Tonia L. Meonske*

NOVEMBER 30, 2006